International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Cloud Decision-Support Systems : Security Challenges and Issues

Somayeh Sobati Moghadam*
Computer and Electrical Engineering Faculty, Hakim Sabzevari University, Sabzevar, Iran

ABSTRACT

Decision support system is a specific class of information systems to support data-oriented analyses and business performance enhancement. Cloud-based decision support system becomes a popular choice because of the value it can provide to the businesses. However, since decision support data are very sensitive, data privacy remains one of the top concerns. In this paper, we review the security and cryptographic mechanisms that aim at making decision support system secure in a cloud environment, and discuss current related research challenges.
Keywords : Decision Support System, Cloud Computing, Data Privacy, Cloud Security.

I. INTRODUCTION

Cloud computing offers a variety of services through a pay-per-use model on the Internet. The flexibility cloud computing offers is very appealing for many organizations, especially mid-sized and small ones, because it provides reduced start-up costs and means to financially cope with variations in system usage. Outsourcing data to the cloud is particularly interesting [34]. However, decision-support data are particularly sensible, e.g., personal data, health-related data, business data. Data warehouses (DWs) is a read-only analytical database is the foundation of decision support system [42]. Traditionally, decision support data are stored in central repositories, i.e., data warehouses (DWs), which consolidate historical data from different sources and allow on-line analysis processing (OLAP). Outsourcing decision-support data in the cloud raises security issues. With increasingly sophisticated internal and external cloud attacks, traditional security mechanisms are no longer sufficient to protect such data [36].

In this paper, we review the security mechanisms that may be implemented to protect the security of cloud decision-support data. We classify such security mechanisms in three classes:
differential privacy (Section 2), threshold cryptography (Section 3) and cryptographic schemes allowing computations over encrypted data (Section4). We discuss the various challenges in this research area as well as implementation barriers in Section 5. Finally, Section 6 concludes the paper.

II. DIFFERENTIAL PRIVACY

Confidential data are normally not directly available. Yet, authorized users may combine query answers with external background knowledge to infer sensitive information. Such threats are known as linkage and probabilistic attacks. To preserve privacy, data must be sanitized by removing well-known identifiers such as names or social security numbers and using data anonymization techniques such as perturbation and masking [11].

Data masking simply replaces original data. Some database management systems (DBMSs) natively include such features [6]. Obscuring query processing and results may be achieved either by rejecting queries leading to privacy disclosure or through methods such as k-anonymity, l-diversity, and t-closeness, which introduce noise in data. However, such techniques must be implemented very carefully to keep a good balance between privacy and utility of the outcome [22].

In the context of DWs, privacy-preserving OLAP in distributed environment can be achieved by table perturbation and reconstruction [1]. Algorithms can reconstruct count aggregates over sub-cubes.

Unfortunately, other aggregation functions such as sum and minimum are not supported. Adding noise to query answer is used, too [33], with several techniques automatically adjusting the scale of noise to reduce relative errors while still ensuring differential privacy.

Finally, (k, e)-anonymity helps anonymize numerical attributes [26], i.e., the observed measures or KPIs in DWs. Partition and permutation are used to preserve privacy when aggregating data from large databases. However, although such a perturbation approach prevents privacy breaches, it can also lead to errors in aggregation results.

III. THRESHOLD CRYPTOGRAPHY

Threshold cryptography is a mechanism in which one person or authority alone cannot access data, whereas a group of authorities can, under some conditions. Moreover, secret data must be stored in several locations to avoid presenting a single point of attack.

The basis of threshold cryptosystems was introduced by Shamir in 1979 [27]. In Shamir's secret sharing, the secret is mathematically divided into pieces that are stored at n participants', with $t \leq n$ participants being required to reconstruct the secret. Secret sharing is very flexible and scalable, and one can easily imagine participants being cloud service providers (CSPs). The first actual threshold encryption scheme couples an ElGamal cryptosystem with secret sharing [10]. It has been enhanced with signatures to guarantees verifiability and robustness [7], [28].

Several schemes exploit threshold encryption in a distributed data management context. [29]'s protocol allows participants to collaboratively compute aggregation queries without gaining knowledge of intermediate results. Moreover, users can verify query results with the help of signatures. However, this protocol is strictly limited to a specific kind of databases with particular schemas, and only allows sum and average aggregations. Other systems allow exact match and range queries, as well as updates, given index keys as predicates [32]. However, aggregate queries are not addressed.

IV. COMPUTATION OVER ENCRYPTED DATA

1) Suitable Cryptographic Schemes

Homomorphic Cryptography: Most encryption schemes require data to be decrypted before they can be processed. When processing data in the cloud, this would mean the CSP has full access to data, which is unacceptable. Only homomorphic encryption (HE) allows performing arbitrary computation over encrypted data without decryption [22].

Fully homomorphic encryption (FHE) allow implementing any function over encrypted data [13]. However, it requires so much computing power that it cannot be used in practice. Even though many improvements have been proposed for, e.g., reducing encryption key size or eliminating [13]'s bootstrapping procedure, building an efficient and usable FHE is still a challenge. Yet, somewhat homomorphic encryption (SWHE), such as the Paillier encryption scheme [24], can be used in practice, by allowing only a certain number of operations over encrypted data. The HELib1 library indeed proposes many implementations and optimizations of SWHEs.

Functional Encryption: In recent years, functional encryption (FE) has emerged as a new paradigm in cryptography [23]. FE is a public key encryption system that supports "partial" decryption keys. The data owner creates a secret key SK_f for any function f, and any user can efficiently compute f(x) over encrypted data knowing SK_f [14]. Decrypting cipher $c = E(pk, m)$ using SK_f reveals f(m) and nothing else [4].

Traditional public-key encryption is actually a specific case of FE. For example, in identity or attribute-based encryption, SK_f is a user identifier or a set of attributes [5]. Finally, although FE seems similar to FHE in terms of functionality, a crucial difference is that FHE outputs encrypted results, while the output of FE is available in clear [5].

When outsourcing data to the cloud, FE offers an elegant solution for, e.g., spam filtering on encrypted emails, mining big data or delegating computation to multiple clients, because it is non-interactive. However, implementation barriers are still a great challenge.

2) Querying Encrypted Data

Comparison: Comparison over encrypted data can be evaluated through either secure multiparty computation (SMC) or order preserving encryption (OPE). SMC is based on Yao's millionaire problem [35], [41] which implements a protocol for "greater than" comparison of two private values [8]. Although SMC approaches are secure, they cannot currently be implemented.

OPE enables to perform range queries over encrypted data while preserving the order of clear text in ciphertext [19]. OPE is a weak encryption scheme, because it reveals order, and is thus vulnerable to plain-text attacks [19], [25]. If an attacker has knowledge about the distribution patterns of plain texts, s/he can map them to discover the encryption key or infer encrypted values.

To overcome this drawback, multivalued OPE (MVOPE) encrypts the same data to different ciphertexts [18], [20], while preserving in cipher texts the order of unencrypted values. MV-OPE helps to compute operations such as $=, <, >, \leq, \geq, \neq,$ min; max and count [20].

Search: Searching over encrypted data is an important problem because it is the primary solution to access outsourced data stored at an untrusted CSP's. Deterministic Encryption is the simplest solution to search over encrypted records in a database. For instance, the Encrypt-with-

Hash deterministic scheme [3] supports fast search on encrypted data. However, it leaks some information to the server. Other approaches provide do more security guarantees, but at the cost of the slower search.

SQL Querying: Bucketization allows executing SQL queries over encrypted data without decryption. Bucketization divides plain text space into buckets, by using the database's distributional properties [17]. Each bucket has an ID and a minimum and maximum value. For each data item in the bucket, the ID is set as a tag. Client queries are mapped to bucket-level queries before being sent to the CSP. They are then evaluated using only the information in the index tags corresponding to data items. Bucketization provides an approximate query processing mechanism. It may indeed return numerous false positives. Thus, results must be post-processed to the client's.

[17] proposed a bucketization procedure for answering multidimensional range queries on multidimensional data, which computes secure indexing tags of data to prevent the server from learning exact values. Bucketization is then considered as an optimization problem to minimize the risk of disclosure. A threshold is defined to help the data owner control the tradeoff between disclosure risk and cost.

3) Secure Data Management Systems

Building a secure DBMS, although much desirable in our cloud DW scenario, is little addressed in the literature. Some solutions have nonetheless been proposed. The most important challenge of secure DBMSs is to formally guarantee privacy, which has not been achieved by the systems we review in this section.

Bucketization-based Approach: [15] developed techniques to allow the bulk of SQL execution being run at the CSP's, with the help of a kind of index that partitions an SQL query. The rest of query needs decryption, and thus is executed at the client's. An algebraic framework is proposed minimized query processing by the client. This approach allows any kind of SQL queries, including aggregation queries, joins, and grouping. However, it requires interaction with the client to complete any query. Row filtering is performed at the CSP's [31].

This approach was extended to support aggregation (sum, count, average, minimum and maximum) on encrypted data without decryption [16]. Formal techniques transform SQL aggregation queries, which are divided into two main categories: certain queries and maybe queries. Certain queries can aggregate data at the CSP's, while maybe query results must be transmitted back to the client to compute final results. Unfortunately, it has been demonstrated that this approach is vulnerable to basic cipher text-only attacks [21].

CryptDB and MONOMI: CryptDB aims to protect confidentiality and manages query processing over encrypted data [25]. CryptDB mostly uses symmetric key encryption and introduces a particular encryption scheme for any given data item, based on queries observed at run-time. Moreover, CryptDB uses chain encryption keys for passwords. As a result, if a user is

not logged into the application, an adversary cannot decrypt the users's data. Encryption in CryptDB is like onion layers that store multiple ciphertexts within each other. The outermost layers provide maximum security, whereas inner layers provide more functionality and less security.

To execute the queries over encrypted data, CryptDB uses a trusted proxy server that rewrites queries before sending them to the CSP. After executing queries at the CSP's, the results are sent back to the proxy server, which decrypts the result and sends the final un-encrypted results to the user's application.

MONOMI builds upon CryptDB to allow analytical queries over encrypted data outsourced to the cloud [31]. MONOMI allows many more operations than previous systems, i.e., 19 out of the 22 queries of the TPC-H decision-support benchmark [30], while [15] and CrypDB supported 2 and 4 queries, respectively. Moreover, MONOMI addresses performance issues caused by large DW volumes and slow query processing over encrypted data, e.g., by splitting queries.

However, the drawback of MONIMI is the heavy communication load between client and server. For instance, intermediate results may be exchanged several times to execute the different parts of a split query [31].

SDB: [40] introduced a secure query processing system, SDB, using data interoperability that allows a wide range of complex queries to be computed by the CSP. By interoperability, the output of an operator is used as input of another operator. SDB implements the Secure Multiparty Computation (SMC) model [41], but unlike SMC, a secret value is split into two shares, one is stored at the CSP's and another at the user's. SDB supports a wide range of SQL queries (e.g., all TPC-H queries [30]) efficiently.

Column-oriented DBMS: [12] conducted a comprehensive study on answering sum and average aggregation queries in data outsourcing models. They demonstrate that the performance of an HE scheme designed in a novel way is comparable to that of traditional, symmetric encryption scheme (e.g., DES), in which computation is performed on plaintext after decryption. The proposed HE scheme operates on a much larger block size than single numeric values, which helps manipulate multiple data values in large encryption blocks. The interesting point in this work is that a column-oriented DBMS, which is the most appropriate model for DW storage [9], is used. This scheme can handle aggregate queries with indexes, as well as grouping queries. SQL HAVING clauses are not supported, though.

4) Secure DWs

ABACUS: [38] proposed a solution based on Shamir's secret sharing [27] to preserve the privacy of distributed DWs. A middleware called ABACUS was developed which allow the execution of queries among distributed DWs. ABACUS allows performing intersection, join, and aggregation queries in a privacy-preserving manner. The middleware operates as a proxy, which allows authorized users querying multiple private DWs. One-way hash functions,e.g., SHA-1, MD4, and MD5 [39], are used to handle join and intersections. Shamir's secret sharing is implemented to perform SUM and AVERAGE. Analytical evaluations confirm the efficiency and scalability of ABACUS.

MOBAT: [37] introduced a data masking technique for DWs which provides a trade-off between privacy and performance. A middleware called MOBAT is set between the user and the server. MOBAT has access to some encrypted metadata such as private keys, rewrites the queries, and obscures data before storing at the server. Each numerical value is masked using three keys, among them two are private and one is public and stored along masked data at the server. For each value in row j in a column, a public key $k_{3,j}$ is stored along with the masked value at the server. Two private keys k_1 and $k_{2,j}$ for j^{th} column are encrypted and stored at the server, too. Experimental evaluations show reduced computational and storage overhead of the proposed masking technique compared with encryption-based solutions.

Partitioned encrypted DWs: [20] proposed a novel method for encrypting and querying a DW hosted in the cloud. This scheme generates indistinguishable encrypted data, allowing the execution of multidimensional queries over the encrypted DW. The DW is horizontally partitioned among several DBMSs. A master DW maintains the address of each partition. On the client side, a secure host is responsible for

encrypting query parameters and for decrypting results. Sum aggregations and data grouping must be computed by the secure host after decryption. To allow grouping on encrypted data, two different multivalued encryption schemes help preserve order in encrypted data, but they do not support minimum and maximum aggregation queries and induce a heavy overhead.

fVSS: [2] proposed a novel approach for securing cloud DWs by flexible verifiable secret sharing, fVSS. fVSS encrypts and shares data over multiple CSP and allows OLAP queries without reconstruction. This scheme also includes inner and outer data verification to check data correctness and the honesty of participants. Moreover, fVSS optimizes data volume and thus reduces outsourcing costs in pay-as-you-go model in the cloud.

V. RESEARCH CHALLENGES AND ISSUES

The weaker cryptographic approaches, such as OPE [25], reveal significant information. Thus, any cryptography method that does not meet rigorous cryptography based security standards must be used carefully. When encrypting a DW's multidimensional schema, to what level should be pushed encryption? If all data, including keys, are encrypted, this impairs the processing of joins. But does having primary and foreign keys unencrypted reveal any information about data? OLAP queries commonly involve aggregations over measures. Thus, HE sounds like an appropriate choice for encryption. For instance, Paillier encryption can be used to sum encrypted data, but the cost of decryption at the client's can remain high in some situations. As a result, it can be more efficient to decrypt data at the client's rather than executing aggregation queries over encrypted data at the CSP's [31]. Moreover, HE cannot preserve order in encrypted data. Thus, when sorting, grouping and range operations must be performed, as is common in OLAP, order-preserving encryption schemes must be considered, although they induce a storage overhead that negatively influences performance and cost.

Performance optimization techniques, such as indexing, partitioning or view materialization, can apply onto encrypted data. However, they speed up certain queries but slow down others [31]. As a result, it is crucial to select a cryptographic method that meets all usage constraints. Again, a tradeoff must be defined to meet the intended level of privacy while minimizing the impact on performance. For example, in bucketization, increasing the number of buckets impairs performance, while a smaller number of buckets increases the risk of data disclosure.

In conclusion, the main challenge in secure cloud DW management, which remains timely, can be stated as follows.

"How to choose and implement security mechanisms that overcome computational and storage overheads, while guaranteeing data security in cloud DWs?"

VI. CONCLUSION

In this paper, we reviewed the security mechanisms that can nowadays be in the deployment of cloud decision-support data. We particularly focused on the cryptographic schemes and the (would-be) secure systems that enable executing queries over encrypted data without decryption. This survey highlights their benefits and barriers of existing solutions in a cloud computing context, and hints at future, practical solutions.

VII. REFERENCES

[1]. Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. Privacy preserving olap. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pages 251-262. ACM, 2005.

[2]. Varunya Attasena, Nouria Harbi, and Jerome Darmont. A novel multi-secret sharing approach for secure data warehousing and on-line analysis processing in the cloud. International Journal of Data Warehousing and Mining, 11(2):21-42, April-June 2015.

[3]. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Advances in Cryptology-CRYPTO 2007, pages 535-552. Springer, 2007.

[4]. Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order revealing encryption: Multi-input functional encryption without obfuscation. In Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of

Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, pages 563-594, 2015.

[5]. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. Communications of the ACM, 55(11):56-64, 2012.

[6]. Oracle Corporation. Data masking best practices. Oracle White Paper, 2010.

[7]. Ronald Cramery, Rosario Gennaroz, and Berry Schoenmakersx. A secure and optimally cient multi-authority election scheme.1997.

[8]. Ivan Damgard, Martin Geisler, and Mikkel Kroigaard. Efficient and secure comparison for on-line auctions. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, Information Security and Privacy, volume 4586 of Lecture Notes in Computer Science, pages 416-430. Springer Berlin Heidelberg, 2007.

[9]. Khaled Dehdouh, Fadila Bentayeb, Omar Boussaid, and Nadia Kabachi. Towards an OLAP environment for column-oriented data warehouses. In 16th International Conference on Data Warehousing and Knowledge Discovery, DaWaK 2014, Munich, Germany, Lecture Notes in Computer Science, pages 221-232, 2014.

[10]. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Advances in Cryptology—CRYPTO'89 Proceedings, pages 307-315. Springer, 1990.

[11]. Cynthia Dwork. Differential privacy. In Encyclopedia of Cryptography and Security, pages 338-340. Springer, 2011.

[12]. Tingjian Ge and Stan Zdonik. Answering aggregation queries in a secure system model. In Proceedings of the 33rd international conference on Very large data bases, pages 519-530. VLDB Endowment, 2007.

[13]. Craig Gentry. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009.

[14]. Shafi Goldwasser, S Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Advances in Cryptology-EUROCRYPT 2014, pages 578-602. Springer, 2014.

[15]. Hakan Hacigumu¸s, Bala Iyer, Chen Li, and Sharad Mehrotra. Executing sql over encrypted data in the database-service-provider model. In Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD '02, pages 216-227, New York, NY, USA, ACM, 2002.

[16]. Hakan Hacıgumu¸s, Bala Iyer, and Sharad Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In Database Systems for Advanced Applications, pages 125-136. Springer, 2004.

[17]. Bijit Hore, Sharad Mehrotra, Mustafa Canim, and Murat Kantarcioglu. Secure multidimensional range queries over outsourced data. The VLDB Journal—The International Journal on Very Large Data Bases, 21(3):333-358, 2012.

[18]. Hasan Kadhem, Toshiyuki Amagasa, and Hiroyuki Kitagawa. A secure and efficient order preserving encryption scheme for relational databases. In KMIS, pages 25-35, 2010.

[19]. Hasan Kadhem, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Optimization techniques for range queries in the multivaluedpartial order preserving encryption scheme. In Knowledge Discovery, Knowledge Engineering and Knowledge Management, pages 338-353. Springer, 2013.

[20]. Claudivan Cruz Lopes, Valéria Cesário Times, Stan Matwin, Ricardo Rodrigues Ciferri, and Cristina Dutra de Aguiar Ciferri. Processing olap queries over an encrypted data warehouse stored in the cloud. In Data Warehousing and Knowledge Discovery, pages 195-207. Springer, 2014.

[21]. Einar Mykletun and Gene Tsudik. Aggregation queries in the database-as-a-service model. In Data and Applications Security XX, pages 89-103. Springer, 2006.

[22]. Christian Neuhaus and Andreas Polze. Cloud security mechanisms.

[23]. Adam O'Neill. Definitional issues in functional encryption. IACR Cryptology ePrint Archive, 2010:556, 2010.

[24]. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Advances in cryptology EUROCRYPT'99, pages 223-238. Springer, 1999.

[25]. Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, pages 85-100. ACM, 2011.

[26]. Cecilia M Procopiuc and Divesh Srivastava. Efficient table anonymization for aggregate query answering. In Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, pages 1291-1294. IEEE, 2009.

[27]. Adi Shamir. How to share a secret. Communications of the ACM, 22(11):612-613, 1979.

[28]. Victor Shoup. Practical threshold signatures. In Advances in Cryptology—EUROCRYPT 2000, pages 207-220. Springer, 2000.

[29]. Brian Thompson, Stuart Haber, William G Horne, Tomas Sander, and Danfeng Yao. Privacy-preserving computation and verification of aggregate queries on outsourced databases. In Privacy Enhancing Technologies, pages 185-201. Springer, 2009.

[30]. Transaction Processing Performance Council. TPC Benchmark H Standard Specification Revision 2.8.0. www.tpc.org/tpch/, 2008.

[31]. Stephen Tu, M Frans Kaashoek, Samuel Madden, and Nickolai Zeldovich. Processing analytical queries over encrypted data. In Proceedings of the VLDB Endowment, volume 6, pages 289-300. VLDB Endowment, 2013.

[32]. Shiyuan Wang, Divyakant Agrawal, and Amr El Abbadi. A comprehensive framework for secure query processing on relational data in the cloud. In Secure Data Management, pages 52-69. Springer, 2011.

[33]. Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. ireduct: Differential privacy with reduced relative errors. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pages 229-240. ACM, 2011.

[34]. Li Xiong, Subramanyam Chitti, and Ling Liu. Preserving data privacy in outsourcing data aggregation services. ACM Transactions on Internet Technology (TOIT), 7(3):17, 2007.

[35]. Andrew Chi-Chih Yao. How to generate and exchange secrets. In Foundations of Computer Science, 1986., 27th Annual Symposium on, pages 162-167. IEEE, 1986.

[36]. Noel Yuhanna. Your enterprise database security strategy 2010. Forrester Research, September, 2009.

[37]. Santos, Ricardo Jorge, Jorge Bernardino, and Marco Vieira. "A data masking technique for data warehouses." Proceedings of the 15th Symposium on International Database Engineering & Applications. ACM, 2011.

[38]. Emekci, Fatih, Divyakant Agrawal, and Amr El Abbadi. "Abacus: A distributed middleware for privacy preserving data sharing across private data warehouses." Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware. Springer-Verlag New York, Inc., 2005.

[39]. Secure Hash Standart. http://www.itl.nist.gov/fipspubs/fip180-1.htm.

[40]. Z. He, W. K. Wong, B. Kao, D. W. Cheung, R. Li, S. Yiu, and E. Lo. SDB: A secure query processing system with data interoperability. PVLDB, 8(12):1876-1879, 2015.

[41]. A. C. Yao. Protocols for secure computations. In Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on, pages 160-164. IEEE, 1982.

[42]. Chau, K.W., Cao, Y., Anson, M. and Zhang, J., Application of data warehouse and decision support system in construction anagement. Automation in construction, 12(2), pp.213-224, 2003.